

# Assistent virtual pel suport al desenvolupament dels hàbits digitals en una organització

Víctor Muñoz Molina

## Resum

El projecte consisteix en desenvolupar un Assistent virtual per ajudar a qualsevol organització en millorar els seus hàbits digitals amb el suport de la tecnologia. Es tracta d'un projecte de intel·ligència artificial (IA) basat en tecnologies Microsoft en el que es crearà un Bot amb capacitat d'autoaprenentatge en funció de les preguntes realitzades i el feedback rebut per part dels usuaris de manera que puguin trobar de forma ràpida i precisa les respostes i la informació que estan buscant. L'assistent virtual es desplegarà com un servei al núvol, el qual facilitarà la seva utilització des de plataformes com SharePoint Online, Microsoft Teams o d'altres.

## Paraules clau

Assistent Virtual, hàbits digitals, Bot, núvol, autoaprenentatge, SharePoint, Intel·ligència Artificial, Microsoft

## Abstract

The project will consist of a virtual Assistant to help any organization to improve their digital habits with the support of technology. This is an AI project based on Microsoft technologies that will create a Bot with the capacity for self-learning based on the questions asked and the feedback received from the users so that they can find the answers quickly and the information you are looking for accurately. The virtual assistant will be deployed as a cloud service, which will facilitate its use from platforms such as SharePoint Online, Microsoft Teams or others

## Index Terms

Virtual Assistant, Digital Habits, Bot, Cloud, Self-Learning, SharePoint, Artificial Intelligence, Microsoft

## 1 INTRODUCCIÓ

En els darrers anys hi ha hagut un augment en la utilització de la tecnologia a les empreses per tal d'agilitzar processos antiquats, ineficients o perquè simplement es poden millorar amb l'ajuda de les noves tecnologies que sorgeixen. Aquestes tecnologies neixen amb l'objectiu d'agilitzar processos existents; o també pot passar que amb el naixement de la tecnologia sorgeixi un nou procés que abans no podia fer-se. Fixant-nos en les dades recollides durant el 2017 i el primer trimestre de l'any passat per l'INE (Instituto Nacional de Estadística)[1] trobem que cada cop més empreses fan ús de les noves tecnologies i eines modernes com ara solucions al núvol, aplicacions web, etc. Com podem veure a la Figura 1 hi ha un ús intensiu de les TIC a les empreses i també observem que aquest ús s'intensifica en les empreses amb més de 10 treballadors:

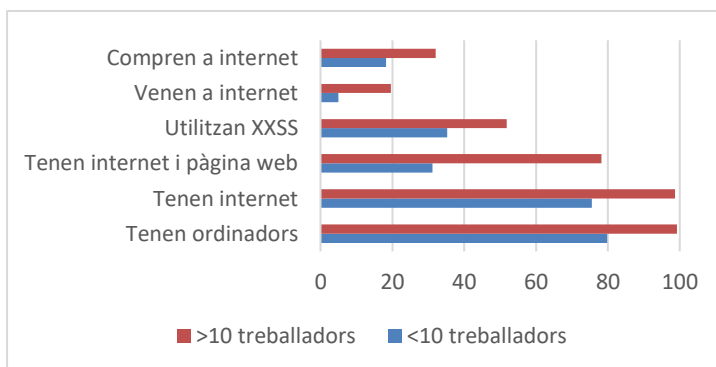


Figura 1 Estadística sobre la utilització de TIC a les empreses

Amb aquestes dades podem extreure que avui en dia les empreses veuen molt important la integració de la tecnologia a les empreses. Això però, no sempre resulta beneficiós, ja que fer-ne un mal ús o un ús pobre pot arribar a ser en alguns casos perjudicial i empitjorar el rendiment de l'Empresa.

Això ha propiciat el desenvolupament de diferents eines per ajudar als usuaris de les noves tecnologies que no saben fer-ne ús, degut a que no han tingut l'oportunitat d'aprendre o bé perquè la tecnologia en qüestió sigui complexa, a aprendre a utilitzar totes aquestes tecnologies adequadament. Alguns exemples d'eines per ajudar als usuaris són els sistemes de FAQs (Frecuent Asked Questions), un document on es recullen les preguntes més freqüents respecte a la tecnologia o altres coses. Un altre exemple d'eina per ajudar als usuaris són els Bots, als quals l'usuari pot interactuar de diferents maneres (per text, amb la veu, amb gestos), i el bot respon d'una manera o d'una altra depenent de l'entrada de l'usuari.

## 2 OBJECTIUS

L'objectiu final d'aquest TFG consisteix en desenvolupar un ChatBot per donar suport a usuaris de SharePoint. SharePoint es un servei al núvol on es poden crear webs on compartir, enmagatzemar i organitzar informació des de diferents dispositius[2]. Aquest ChatBot es desenvoluparà com un servei desplegat al núvol, que sigui capaç de respondre la major part de les preguntes d'un usuari qualsevol amb dubtes per la utilització d'eines Online de Microsoft. La funcionalitat principal de l'eina és aclarir els dubtes que un nou usuari sense experiència prèvia pot tenir.

Aquest ChatBot es pretén crear sobre l'eina Azure Bot Service de Microsoft i integrant-hi els serveis: LUIS (Language Understanding)[3] de Microsoft per afegir comprensió del llenguatge i el servei QnAMaker[4] per realitzar la KB (*Knowledge Base*, base de coneixement)[5] del nostre bot, en altres paraules la KB és un diccionari de les preguntes/respostes del nostre bot.

Per tant, per tal de desenvolupar el ChatBot de suport al núvol, aquest TFG adreçarà 3 punts:

1. Desenvolupar una aplicació de LUIS, que extreurà la intenció de l'usuari.
2. crear una aplicació de QnAMaker, que triarà la resposta adient a la intenció de l'usuari.
3. crear un ChatBot que integri aquestes dues aplicacions.

El ChatBot es desplegarà en un servei a Azure (Cloud de Microsoft) perquè es pugui fer servir des de diferents plataformes de manera fàcil i ràpida.

A continuació a la Figura 2 es veu un exemple de com podria ser la visualització d'un ChatBot a una web amb una interfície bàsica. Una barra per escriure el missatge que es vol enviar i un botó per enviar el missatge i un altre per tancar la finestra del ChatBot.

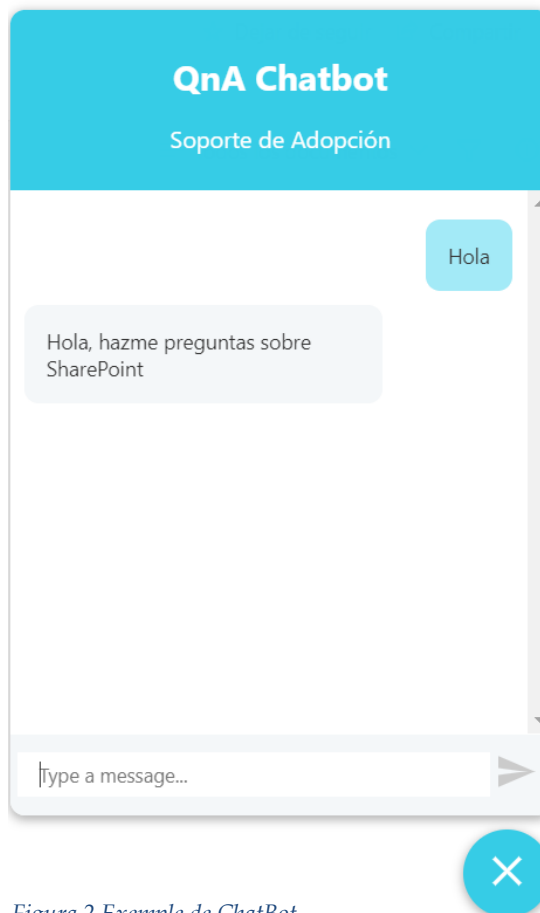


Figura 2 Exemple de ChatBot

## 3 ESTAT DE L'ART : ELS CHATBOTS

Els chatbots són bàsicament interfícies conversacionals amb el propòsit d'ajudar individus amb problemes de tot tipus.

Segurament algun cop t'has trobat amb problemes mentre navegaves a una web i has buscat ajuda a un bot o mentre navegaves a una pàgina s'obria un bot automàticament d'algú que et deia alguna cosa com: "Et puc ajudar en alguna cosa?". Són més dinàmics i útils que els simples FAQs que s'havien utilitzat sempre a les pàgines web per donar suport a l'usuari. Sabem que ja hi ha bots a moltes webs, i que quasi segur que hem vist algun, però, quines limitacions tenen? Fins a on poden arribar? En un futur ens ajudaran aquests tipus de sistemes i desapareixeran altres sistemes antics com l'ajuda d'un dependent, d'un encarregat, d'un comercial, etc? Anem a veure-ho.

- E-mail de contacte: victor.munozmo@e-campus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Dèbora Gil Resina (Computer Vision Center)
- Curs 2018/19

### 3.1 Estat actual

Actualment aquests ChatBots poden respondre en base a diferents criteris, els veurem de més simples a més complexos:

- Basats en un menú o botons: aquesta classe de ChatBots tenen un funcionament molt bàsic i poc flexible. Basats en un menú d'opcions que el bot mostra a l'usuari i aquest escull entre aquesta sèrie d'opcions. Aquest tipus de ChatBot si es fa adequadament no ha de tenir errors ja que l'input de l'usuari està controlat en cada moment, els següents tipus en canvi, poden donar peu a que sorgeixin errors si l'input de l'usuari no s'ha previngut.



Figura 3 Exemple de bot basat en menú

- Basats en reconeixement de paraules clau: aquesta classe de ChatBots a diferència del tipus anterior accepta input de l'usuari i intenta respondre apropiadament, són molt més flexibles que els anteriors però també són més complexos i donen lloc a possibles errors a l'hora d'entendre el que l'usuari diu.

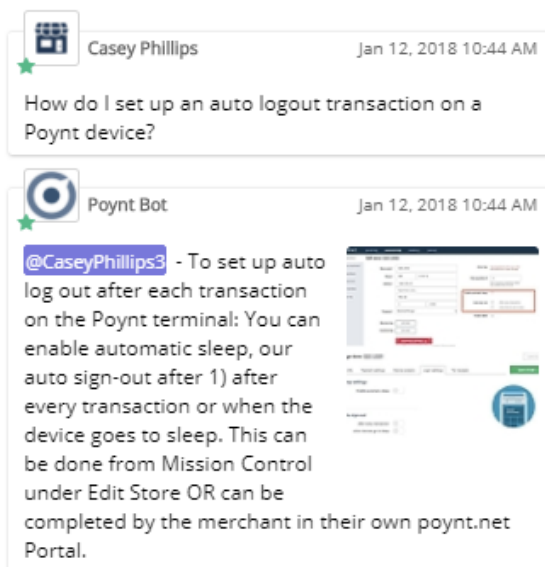


Figura 4 Exemple de bot basat en reconeixement de paraules clau

- Basats en el context: aquest tipus de ChatBot és el més avançat de tots. Són ChatBots que introdueixen l'ús de l'IA i del ML (Machine Learning, Aprenentatge computacional). A diferència dels ChatBots basats en reconeixement de paraules claus, aquests poden millorar sols aprenent a partir de l'input dels usuaris com per exemple: de la manera en que parlen, de què parlen, com ho diuen, etc. Els més avançats poden recordar converses i auto millorar-se contínuament.

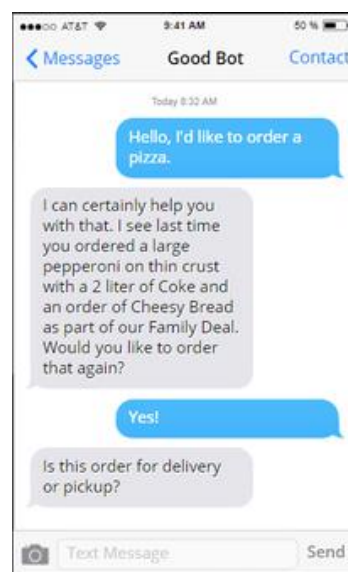


Figura 5 Exemple de bot basat en el contexte

Els més simples disposen d'un arbre de decisions per on l'usuari va movent-se depenent de les opcions que esculli, opcions limitades que li mostra el bot en cada moment. Els segons més simples disposen d'un diccionari de claus - valor, i el funcionament és tan simple com que si l'input que li apliquen al bot és una de les claus que té al diccionari llavors ell retorna el valor. Per exemple si fos un ChatBot d'una botiga online i l'usuari tingués dubtes sobre el procés de pagament de la botiga i més concretament quins tipus de pagament accepten podria escriure al bot "Tipus de pagament" i imaginem que el bot tingués al diccionari l'entrada: {"Tipus de pagament" : "Acceptem: Visa, MasterCard, PayPal i d'altres"}. Llavors el bot li respondria: "Acceptem Visa, MasterCard, PayPal i d'altres". D'altra banda si el bot no tingués cap entrada per l'input de l'usuari podrien haver dos casos, que tingués una sortida predeterminada pels missatges que li entren com a input i que no té guardats al diccionari, com ara: "No t'he entès, intenta dir-m'ho d'una altra manera"; o que no tingués cap missatge establert i simplement no respongués a l'usuari. Hi ha d'altres però que són molt més complexos, i que basen les seves respostes en diccionaris en què els inputs poden tenir diferents respostes alhora, i on primerament es fa un anàlisis del text que l'usuari introdueix per tal d'aclarir el significat que l'usuari té amb la qüestió introduïda per saber quina resposta cal escollir en cada moment. Aquests últims serien del tipus basat en el context, explicat anteriorment.

### 3.2 Limitacions

Les limitacions que tenen ara mateix els ChatBots específicament, són bastant notables, què vol dir això; qualsevol usuari que faci servir en algun moment un bot no trigarà molt en adonar-se que és un bot. Encara estem lluny de que un bot passi el conegut test de Turing[2], que consisteix en que un humà avalués converses entre un humà i un bot, coneixent des d'un principi que existeix un bot a la conversa però sense saber qui és l'humà i qui és la màquina o bot, llavors si després de tenir una conversa entre humà i màquina l'avaluador no sap diferenciar qui era el bot i qui l'humà significa que el bot ha passat la prova, doncs aquesta prova encara està lluny de superar-se satisfactòriament. Fins ara el bot que ha estat més a prop d'aconseguir-ho és un bot anomenat Mitsuku[6], ha sigut 4 vegades guanyador del premi *Loebner Prize Turing Test*. Però tot i així el bot encara es pot detectar en segons quines situacions tot i que és bastant real, [aquí](#)[7] podreu veure les converses que va tenir amb els diferents jutges del premi *Loebner Prize Turing Test*.

### 3.3 Què podem esperar

Dels ChatBots podem esperar una millora contínua al pas dels anys, al nivell en què millorin les AI dedicades a la comprensió del llenguatge natural.

Les grans companyies com ara Google, Amazon, Microsoft, etc. Cada cop construeixen més tecnologia basada en xarxes neuronals que permeten que sistemes com els ChatBot millorin a l'hora d'entendre i interactuar amb persones. Un cop sigui més comú i normal el desenvolupament de xarxes neuronals els ChatBots començaran a superar el

Test de Turing, la seva presència a Internet serà més prolífera i alguns experts asseguren que en el futur estaran presents a molts aspectes de la nostra vida com ara als cercadors com Google que en comptes de tenir una barra de cerca funcionaran amb ChatBots i la barra de cerca serà història.

## 4 METODOLOGIA

L'aplicació es crearà utilitzant l'eina Azure Bot Service de Microsoft, és una eina que facilita la creació de bots mitjançant llibreries de programació amb elements que s'utilitzen en el desenvolupament de chatbots, el llenguatge que s'utilitzarà serà C#, que facilita la tasca pels programadors amb experiència en desenvolupament d'aplicacions o serveis de .NET. Sobre aquest framework treballarem també amb LUIS, l'eina de Microsoft per entendre el llenguatge. Aquesta eina ajuda a entendre la intencionalitat que hi ha a una frase per fer més fàcil el procés de resposta del bot i sobretot fer-lo més precís. LUIS ens ajuda a entendre el context, la intenció i els objectes als que es refereix l'usuari en el seu input. També treballarem amb l'eina QnAMaker. Aquesta eina ens ajuda a realitzar un diccionari de preguntes-respostes perquè sigui la base de preguntes que el nostre ChatBot pot respondre. El desenvolupament està dividit en 3 parts interconnectades:

- L'app de LUIS
- L'app de QnAMaker
- El servei del ChatBot

## 5. DESENVOLUPAMENT

Els tres objectius principals d'aquest TFG són crear les aplicacions de LUIS, de QnAMaker i el servei del ChatBot. Les dues aplicacions estaran unides pel servei del ChatBot. Aquest el que farà serà agafar l'input de l'usuari i el passarà a l'app de LUIS. LUIS farà la interpretació de l'input de l'usuari i li retornarà al servei del ChatBot. Aquest després li passarà a l'aplicació de QnAMaker i QnAMaker respon amb el valor que correspon amb l'output de LUIS al servei del ChatBot que mostrarà aquest valor a l'usuari.

A la següent figura podem observar un esquema del funcionament del ChatBot des de que un usuari introdueix un input fins que rep la resposta. El procés és el següent:

1. Usuari introdueix input al ChatBot.
2. El ChatBot passa l'input a l'app de LUIS.
3. L'app de LUIS retorna la resposta al ChatBot.
4. El ChatBot envia a l'app de QnAMaker la sortida de l'aplicació de LUIS.
5. L'aplicació de QnAMaker envia la resposta adequada al ChatBot.
6. El ChatBot rep la resposta de QnAMaker i li mostra al usuari per pantalla.

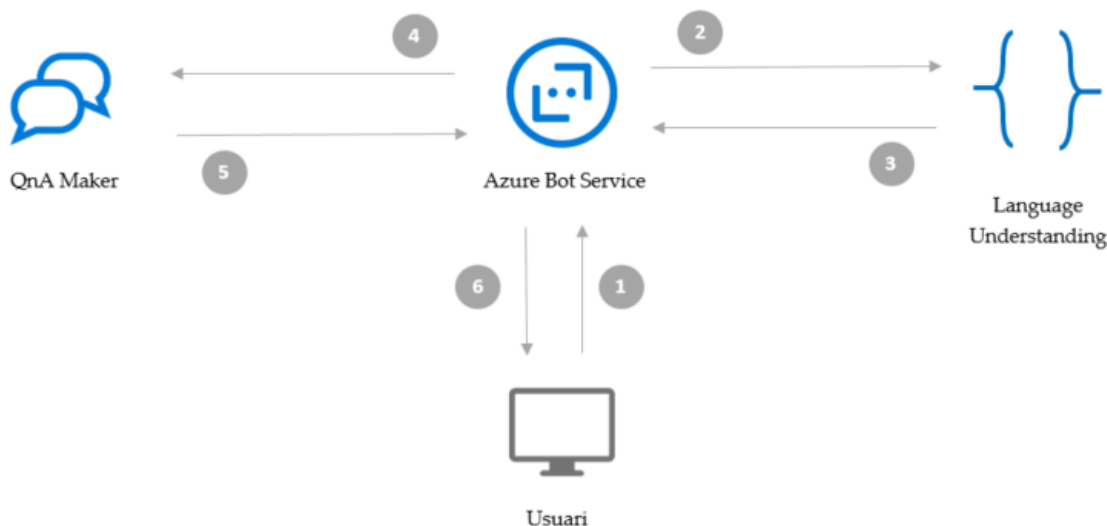


Figura 6 Funcionament del bot des de que un usuari introdueix un input fins que li arriba la resposta.

## 5.1 L'app de LUIS

L'aplicació de LUIS el que farà és traduir el missatge de l'usuari a una frase amb la intencionalitat bàsica de l'usuari. Per exemple passarem de "Como podría subir muchos documentos a una lista de SharePoint" → "Subir documentos lista". Aquesta aplicació és essencial per donar-li flexibilitat al ChatBot i ampliar el seu potencial. El seu funcionament és simple si es vol fer una aplicació que no sigui molt complexa com la que podem realitzar en aquest TFG, utilitzarem dos elements bàsics de LUIS:

- Intencions: seria la tasca o acció que l'usuari vol realitzar, és una finalitat o un objectiu que expressa l'usuari.[8]
- Entities: són entitats que ens serveixen per l'extracció de dades de l'input de l'usuari, tenen relació amb la finalitat o objectiu que expressa l'usuari.[9]

Seguint amb l'exemple anterior "Cómo podría subir muchos documentos a una lista de SharePoint" la intenció seria "subir" i les entitats serien "documentos" i "lista". Amb això l'aplicació extreu la intencionalitat que hem comentat abans, "subir documentos lista".

Per a la implementació de la nostra aplicació de LUIS ens cal seguir 3 passos principals:

1. Identificació de les intencions i entitats que tindrem i declarar-les.
2. Entrenar el model amb les intencions i entitats declarades.
3. Testejar el model i publicar-lo al núvol.

### 5.1.1 Funcionament de LUIS I: Declaració d'Intencions i Entitats

Primer de tot el que hem de realitzar és la declaració de totes les Intencions i entitats que esperem que el nostre usuari faci servir. En aquest cas hem utilitzat aquests:

- Intencions: *Acceder, Añadir, Comprobar, Crear,*

*Eliminar, Exportar, Modificar, None, Ocultar, Ordenar, Proteger, Saber, Saludar, Subir, Ver.*

- Entities: *Aplicación, Biblioteca de documentos, Carpeta, Columna, Configuración, Contenido del sitio, Desprotegido, Documento, Documentos, Elemento, Excel, Filtro, Flujo, List, Navegacion, No Visible, Pagina, Permisos, Protegido, Sitio, Tipo de contenido, Usuario, Visible, WebPart.*

### 5.1.2 Funcionament de LUIS II: Entrenar el model i testejar-lo

El següent que hauré de fer serà entrenar el model d'IA de LUIS amb els valors declarats. El model que entrena LUIS per darrere és un model NLP, Natural Language Processing probabilístic. Un model d'IA per entendre el llenguatge natural a base de probabilitats, aquest model agafa les intencions i entitats declarades i les utilitza com el conjunt d'entrenament del model. Amb aquest conjunt d'entrenament el model entrenarà un classificador amb puntuacions, aquestes puntuacions dependran de les probabilitats que tenen les entitats i les intencions declarades un cop s'ha entrenat el model. Amb aquestes puntuacions calcularà una sèrie de "punts" per a cada intenció i entitat amb l'input de l'usuari i retornarà en format JSON la intenció amb més puntuació i les entitats identificades a l'input. Cal especificar que no tenim conjunt de testeig, podem testejar-lo nosaltres introduint-li inputs amb una interfície de testeig que ens mostra els resultats però conjunt de testeig com a altres models de IA no n'hi ha. Podríem dir que cada input de l'usuari és un test en sí mateix. Si aquest input el nostre model no el sap classificar el guardarà perquè el puguem classificar nosaltres manualment a posteriori i el model aprèn d'aquesta classificació manual per a futurs inputs semblants, a continuació s'explicarà en detall.

Quan es fa una consulta al nostre model el que realment succeeix és una petició HTTP al servei d'Azure on el nostre model de LUIS està emmagatzemat i ens retorna unes dades en format JSON que contenen la intenció amb més



puntuació del model, és a dir, de totes les intencions possibles la que el model ha extret que es més possible que l'usuari s'estigui referint; i les entitats que el model ha sigut capaç d'identificar. Rarament el model extreu una sola intenció, el que fa com he dit abans, és retornar la intenció que creu que és a la que fa més referència l'usuari. En el següent exemple [Figura 7] podrem veure que té una intenció (*Intent*) de "HRContact" amb una *score* de 0.921233 això significa que els 0.078767 "punts" restants estan repartits entre la resta d'intencions que aquesta aplicació té definides. Això serveix perquè quan hi ha inputs que el model treu una baixa puntuació, això pot passar si no estan ben definides les intencions i un input pot referir-se a més d'una intenció, la resposta contindrà potser una intenció amb una puntuació de 0.192012, amb aquestes puntuacions l'aplicació de LUIS automàticament es guarda aquest input i més endavant a la seva interfície de configuració es podrà repassar aquest input i assignar manualment a quina intenció es referia l'usuari. D'aquesta manera es pot fer una mena "d'aprenentatge supervisat" per a que més endavant l'aplicació de LUIS respongui millor a inputs similars.

A continuació hi ha l'exemple de resposta de l'aplicació de LUIS a l'input "I want to call my HR rep." esmentat abans, en el format que ho retorna (JSON).

```

Language Understanding (LUIS)

{
  "query": "I want to call my HR rep.",
  "topScoringIntent": {
    "intent": "HRContact",
    "score": 0.921233
  },
  "entities": [
    {
      "entity": "call",
      "type": "Contact Type",
      "startIndex": 10,
      "endIndex": 13,
      "score": 0.7615982
    }
  ]
}

```

Figura 7 Exemple de resposta en format JSON de l'input: "I want to call my HR rep."

En aquest exemple el model ha extret que la intenció es contactar amb una persona de RRHH i només ha detectat una entity, "call" que ens dóna informació sobre la intenció de l'usuari, gràcies a aquesta resposta podem entendre quin és l'objectiu de l'usuari i informació per arribar a aquest objectiu. D'aquest JSON extraïem la intenció comentada anteriorment i que ho vol fer mitjançant una trucada.

Un cop podem extreure aquesta informació de l'input de l'usuari podem passar a crear una KB per respondre a l'usuari les diferents intencions o peticions que ens pot fer i que l'aplicació de LUIS interpreta.

### 5.1.3 Funcionament de LUIS III: Publicació del model entrenat

Un últim pas que s'hauria de fer perquè l'aplicació estigui disponible és publicar-la, això només és un tràmit però necessari, el que fa és allotjar aquest model que hem entrenat i testejat al núvol i ens dóna informació de l'endpoint a on hauré de fer les peticions d'HTTP per que vagin dirigides al nostre model.

## 5.2 L'app de QnAMaker

L'aplicació de QnAMaker el que realitzarà serà tenir parelles de preguntes-respostes de manera que li farem arribar la sortida de LUIS, de manera estructurada, perquè sigui capaç de respondre a l'usuari amb el que està demanant. Per això usarem l'element principal de QnAMaker KB, que ens permet guardar parelles clau-valor que utilitzarem per escollir la resposta més adient depenent del input de l'usuari.

KB (Knowledge-Base): una base de dades o diccionari on guardem parelles de strings bàsicament. Seran parelles de clau-valor, de manera que si l'output de l'app de LUIS és una de les claus que tenim a la nostra KB l'output de l'app de QnAMaker serà el valor associat a aquesta clau.

Per la implementació de l'aplicació de QnAMaker necessitarem seguir 3 passos principals:

1. Definir les parelles de clau-valor que s'utilitzaran per entrenar el model.
2. Entrenar el model amb les parelles definides.
3. Testejar el model i publicar-lo al núvol.

El model d'IA que entrena també és un NLP, per classificar l'input que li arriba per poder respondre una pregunta feta de diferents formes tot i que no és tan efectiu com LUIS, que és un servei dedicat només a entendre la intenció de l'usuari, per això es va decidir implementar aquest ChatBot amb les dues aplicacions, tant QnAMaker com LUIS.

### 5.2.1 Funcionament de QnAMaker I: Declaració de les parelles clau-valor

El primer que hem de fer a la nostra aplicació de QnAMaker serà crear les parelles clau-valor/pregunta-resposta que necessitem. Aquestes dependran de quines preguntes creiem que l'usuari farà. Les preguntes de la nostra KB serà el conjunt de training del nostre model, i com passava amb LUIS, no hi ha conjunt de test més enllà dels inputs que l'usuari fa o si fem proves nosaltres abans de publicar el servei al núvol per millorar el rendiment de la nostra aplicació. Com he explicat anteriorment si el servei que es-

tem fent volem que doni resposta a dubtes sobre SharePoint en concret prepararem una KB per preguntes sobre SharePoint. QnAMaker té un punt fort, i és que es poden exportar i importar les KB. Per importar-les accepta més d'un format i a l'hora d'exportar-les crea un arxiu .tsv. Això ens interessarà molt a l'hora d'expandir i fer més gran el nostre ChatBot ja que si per exemple, volem crear un ChatBot de suport a totes les eines d'O365 no haurem de tornar a crear una KB amb preguntes de SharePoint si no que podrem importar la que crearem per aquest ChatBot.

### 5.2.2 Funcionament de QnAMaker II: Entrenar el model i testejar-lo

Molt semblant a com hem fet abans amb LUIS, entrenarem el model de QnAMaker i el testejarem per veure que tot funciona bé. Per testejar-lo l'input que introduïrem ha de ser com l'input que li arribaria de l'app de LUIS. En aquest cas l'output de l'aplicació es una string. Tot i que la sortida és una string la configuració de la resposta (del parell pregunta-resposta) permet configurar la resposta donant-li una mica de format, però és limitat. Les respostes de la Knowledge-Base accepten llenguatge Markdown que permet un cert grau de formateig al text però no és tant com podria ser html per exemple; però això ja et permet introduir enllaços, posar el text en cursiva, entre d'altres opcions. Un exemple de prova seguint amb l'exemple de LUIS de la Figura 6 seria: "HRContact call". Llavors a la nostra KB tindríem per exemple un parell clau-valor on la clau seria: "HRContact call". I el valor seria: "Your HR rep is Jhon Smith [jhons@contoso.com](mailto:jhons@contoso.com) 212-555-1212." Llavors l'output de la nostra aplicació de QnAMaker seria el valor d'aquest parell pregunta-resposta.

### 5.2.3 Funcionament de QnAMaker III: Publicar el model entrenat

Per últim, com s'ha comentat també a l'aplicació de LUIS hauríem de publicar el nostre model perquè s'emmagatzemi a Azure i la interfície ens proveeixi del Endpoint a on hem de fer les peticions HTTP com a LUIS.

## 5.3 El servei del ChatBot

Un cop tenim les aplicacions de LUIS i QnAMaker desenvolupades, testejades i publicades podem realitzar la part final del projecte, el ChatBot. El ChatBot és un servei que s'allotjarà a Azure i que el que farà serà relacionar l'aplicació de LUIS i l'aplicació de QnAMaker. El seu funcionament és molt simple :

1. Recull l'input de l'usuari a la interfície del servei del ChatBot.
2. Envia aquest input a l'endpoint de l'aplicació de LUIS fent una petició HTTP.
3. L'aplicació de LUIS respon al ChatBot, un cop ha passat l'input de l'usuari pel model entrenat prèviament; amb la resposta del model en format JSON.
4. El ChatBot agafa aquest JSON i el passa a una

string extraient la informació rellevant i crea un string seguint el següent esquema: "Intenció Entitat\_1 Entitat\_2 Entitat\_N"

5. Aquest string l'envia, mitjançant una petició HTTP, a l'endpoint de l'aplicació de QnAMaker.
6. L'app de QnAMaker passa l'string pel seu model i extreu a quina clau s'està referint l'input que li ha arribat.
7. L'app de QnAMaker respon al ChatBot amb la resposta corresponent a la pregunta del KB.
8. El ChatBot rep la resposta de l'aplicació de QnAMaker i la mostra per la interfície a l'usuari finalitzant així el procés de resolució d'una qüestió.

### 5.3.1 Funcionament del ChatBot

Per que el ChatBot funcioni com hem explicat anteriorment s'haurà de connectar mitjançant peticions HTTP a les aplicacions de LUIS i QnAMaker. Per fer-ho haurem de programar en C# l'intercanvi de missatges entre el ChatBot i LUIS per una banda i d'altra banda els missatges entre el ChatBot i QnAMaker. Per fer-ho hem utilitzat llibreries que Microsoft posa a la nostra disposició per relacionar aquests dos tipus de serveis utilitzant el seu Bot Framework. En aquest cas s'ha utilitzat la versió 3 del Bot Framework (la versió 4 va sortir a finals de l'any passat). A banda d'altres més genèriques les llibreries específiques més importants i imprescindibles que s'han utilitzat són:

- LUIS: `Microsoft.Bot.Builder.Luis`.
- QnAMaker: `Microsoft.Bot.Builder.Dialogs`.

Aquestes llibreries ens han permès establir les connexions HTTP amb cada aplicació i extreure la informació pertinent de cada resposta.

Haurem d'utilitzar aquestes llibreries per crear un objecte de tipus `QnAMakerService`, on hi especificarem l'endpoint on s'ha de fer la petició HTTP, el domini de l'aplicació i l'ID de la KB feta. També especificarem un objecte de tipus `LuisService` on especificarem l'endpoint per les peticions HTTP, l'ID de l'aplicació de LUIS i el domini a on és l'aplicació. Aquests dos tipus d'objectes: `LuisService` i `QnAMakerService` els definirem al codi amb les propietats i mètodes necessaris pel seu funcionament. Després realitzarem un bloc de programació que definirà un "mètode de diàleg" per cada intenció definida a l'aplicació de LUIS, això farà que depenent de la intenció que retorni LUIS s'executi un codi o un altre, per defecte a tots es el mateix, ja que a totes les intencions volem que enviïn la resposta a QnAMaker un cop s'ha extret la informació del JSON de LUIS i després s'envia la resposta de QnAMaker al ChatBot.

## 6 RESULTATS

Els resultats que hem obtingut de la realització d'aquest TFG es un servei de suport a nous usuaris a l'eina SharePoint ja que aquest servei està allotjat a Azure(Cloud de Microsoft) pot ser instal·lat a qualsevol Web; sigui una web de SharePoint o no. Aquest servei reconeix el que l'usuari vol dir en la majoria dels casos i respon de manera correcta a les qüestions que realitza l'usuari en qüestió. A continuació posaré un parell d'exemples de la versió final del ChatBot en acció, primer especificaré l'input del bot, després la resposta que dona l'aplicació de LUIS un cop s'ha extret la informació del JSON, la parella pregunta-resposta de la KB de QnAMaker i finalment una imatge del que veu l'usuari:

1. Pregunta: "Como subir documentos a una lista de SharePoint"
  - a. Resposta LUIS: "Subir documentos List"
  - b. Parella pregunta-resposta de la KB de QnAMaker, Figura 8. Les frases de l'esquerra són les claus de la parella i la frase de la dreta és la resposta corresponent a aquestes claus.
  - c. Interfície del ChatBot, experiència d'usuari, Figura 9. Tal i com veu l'usuari el resultat final del servei del ChatBot al fer-lo servir.



Question	Answer
<div> <div>^ Source: Editorial</div> <div> <div>subir documento ✕</div> <div>subir documentos ✕</div> <div>subir carpeta ✕</div> <div>subir elemento ✕</div> <div>+ Add alternative phrasing</div> </div> </div>	
	<div> <div>Para subir documentos y/o carpetas puedes mirar [esta página](https://support.office.com/es-es/article/cargar-archivos-o-una-carpeta-en-una-biblioteca-de-documentos-eb18fcba-c953-4d45-8d90-8da66edeacdb) que te explica diferentes formas de subir documentos y/o carpetas a SharePoint.</div> <div>+ Add follow-up prompt</div> </div>

Figura 8 Exemple 1 parella pregunta-resposta QnAMaker

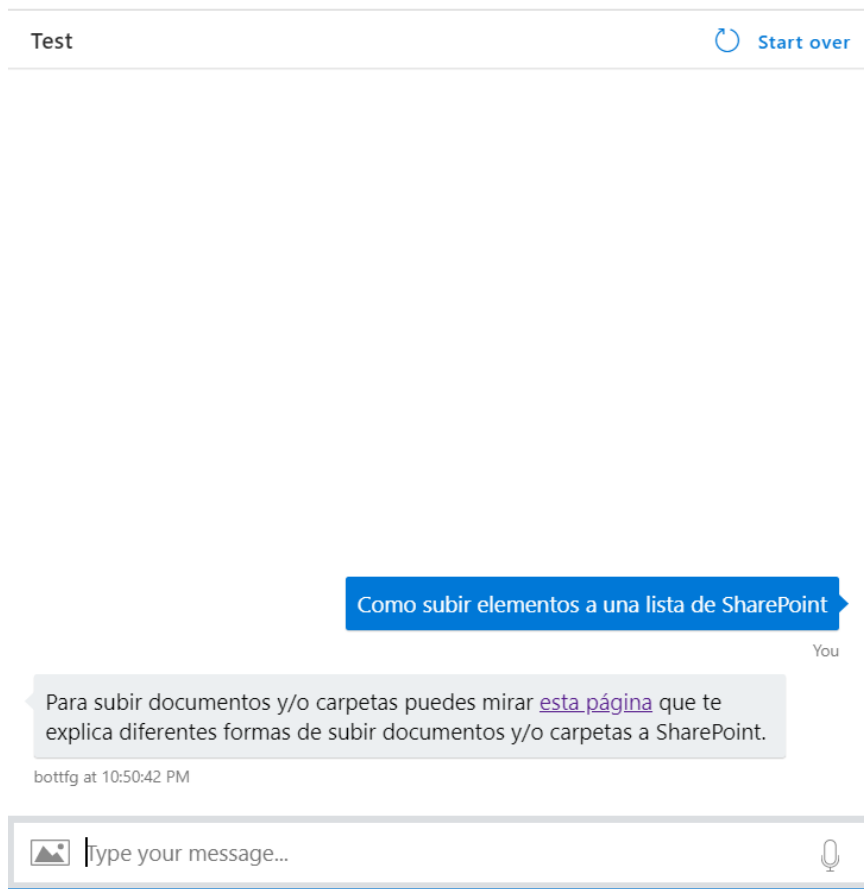


Figura 9 Exemple 1 experiència d'usuari amb l'interfície del ChatBot

2. Pregunta: “Como puedo añadir tipos de contenido a una lista”
  - a. Resposta LUIS: “añadir tipo de contenido List”
  - b. Parella pregunta-resposta QnAMaker, Figura 10.
  - c. Interfície del ChatBot, experiència d’usuari, Figura 11.

Question	Answer
<div> <div> ^ Source: Editorial </div> <div> <div>Añadir tipo de contenido X</div> <div>Añadir tipo de contenido List X</div> <div>Añadir tipo de contenido Sitio X</div> <div>Crear tipo de contenido X</div> <div>Crear tipo de contenido List X</div> <div>Crear tipo de contenido Sitio X</div> <div>+ Add alternative phrasing</div> </div> </div>	
	<div> <div>[Esta página](https://support.office.com/es-es/article/crear-o-personalizar-un-tipo-de-contenido-de-sitio-27eb6551-9867-4201-a819-620c5658a60f) te muestra como crear nuevos tipos de contenido de sitio y en [esta otra] (https://support.office.com/es-es/article/agregar-un-tipo-de-contenido-a-una-lista-o-biblioteca-917366ae-f7a2-47ad-87a5-9689a1884e60) te muestran como añadir los tipos de contenidos creados a la lista o biblioteca que desees.</div> <div>+ Add follow-up prompt</div> </div>

Figura 10 Exemple 2 parella pregunta-resposta de QnAMaker

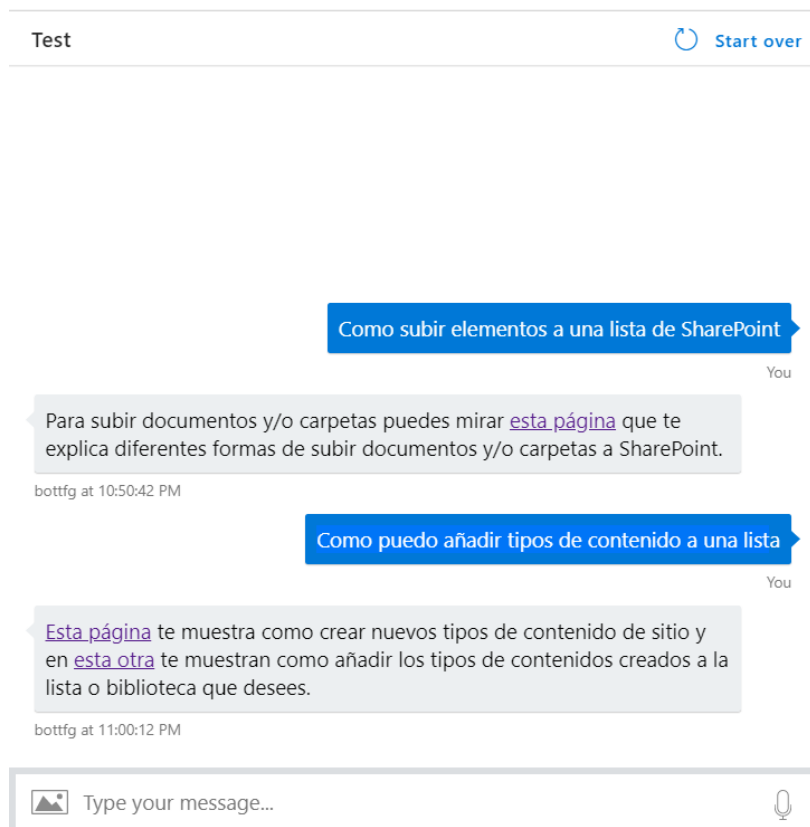


Figura 11 Exemple 2 interfície del ChatBot, experiència d’usuari

El ChatBot l'he testejat amb 5 usuaris, que no tenen coneixements de SharePoint, posant-li 5 activitats bàsiques a realitzar en un lloc web de SharePoint. Les 5 activitats són:

1. Crear una llista de SharePoint.
2. Crear una columna de lloc.
3. Crear un tipus de contingut de lloc.
4. Crear una pàgina web.
5. Afegir la columna i el tipus de contingut creats als punts 2 i 3 a la llista creada al primer punt.

Dels 5 usuaris tots van fer totes 5 activitats a excepció d'un usuari que no va aconseguir fer l'activitat 5. Durant la realització de les activitats van fer-se 31 preguntes entre els 5 usuaris, i d'aquestes 31 el bot va fallar en 4 preguntes, totes les preguntes errònies van ser de l'usuari que no va arribar a fer l'activitat 5. Les preguntes que va fallar el bot van ser a causa d'utilitzar verbs reflexius els quals no havia tingut en compte a l'hora de crear el bot.

També vaig realitzar una prova lliure amb un usuari que té coneixements bàsics mitjans de SharePoint. Va realitzar 29 preguntes y d'aquestes 29 preguntes el bot va respondre bé 23, les 6 restants eren preguntes que no estaven definides com possibles preguntes abans de la prova de l'usuari, amb aquest feedback vaig ampliar les intencions i entitats de l'aplicació de LUIS, la KB de l'aplicació de QnAMaker i el codi del ChatBot per a que les tingues en compte i les respongués en futures conversacions.

## 7 CONCLUSIONS

Podem concloure que hem arribat a l'objectiu principal que era desenvolupar un servei al núvol que donés suport als nous usuaris de l'eina SharePoint. No hi ha cap objectiu que no haguem complert dels que es van marcar inicialment. Podem afegir que gràcies a realitzar aquest projecte he après coneixements molt valuosos sobre les eines de Microsoft, LUIS i QnAMaker; també a desenvolupar aplicacions bot amb Bot Framework i a consolidar coneixements de C#. Per últim destacar que el desenvolupament no s'acaba aquí, tot i que aquest TFG s'ha centrat en el cas d'ús de SharePoint, com s'ha comentat anteriorment, podria donar suport a més eines o temes. Per fer-ho l'únic que s'hauria de modificar són les aplicacions de LUIS i QnAMaker que es realitzaran. En el cas de LUIS canviar totes les variables necessàries pel tipus de llenguatge que serà el seu input; i en el cas de QnAMaker modificar la KB per introduir les preguntes-respostes que hauria de ser capaç de contestar. Un altre tipus de desenvolupament futur que es podria fer seria adaptar el ChatBot amb reconeixement de veu i que puguis interactuar amb comandes per veu amb el ChatBot i a l'inrevés, preparar el ChatBot per contestar per veu, tot i que s'ha de pensar en que amb aquesta funcionalitat les respostes amb hipervincles, o altres formatjos no funcionaran. Aquests serien uns exemples de treball futur per millorar aquest servei de ChatBot.

## 8 AGRAÏMENTS

En aquest apartat m'agradaria agrair a totes les persones que han contribuït a que realitzés aquest treball i fos possible que l'acabés o que simplement m'han ajudat al llarg dels anys a arribar fins aquest punt.

Primer de tot agrair-li a la meua tutora del TFG, Dèbora Gil Resina, pel seu esforç en portar el meu treball i per la seva comprensió amb mi durant la realització d'aquest treball per la alta càrrega de treball que he tingut durant el desenvolupament del mateix.

En segon lloc m'agradaria agrair-li al meu company de feina Ruben Ramos, per ajudar-me en un punt en el que sense ell no sé si hagués pogut continuar el desenvolupament i gràcies a ell vaig poder finalitzar-lo.

Per últim a la meua família que sempre ha estat al meu costat per donar-me suport durant aquesta etapa de la meua vida i sempre, gràcies.

## BIBLIOGRAFIA

- [1] Estadística empreses utilització de les TIC, [Consultat a data 06/03/2019]: [https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica\\_C&cid=1254736176743&menu=ultiDatos&idp=1254735576799](https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176743&menu=ultiDatos&idp=1254735576799)
- [2] Qué es SharePoint? [Consultat a data 10/06/2019]: <https://support.office.com/es-es/article/%C2%BFqu%C3%A9-es-sharepoint-97b915e6-651b-43b2-827d-fb25777f446f>
- [3] LUIS, Language Understanding [Consultat a data 08/03/2019]: <https://www.luis.ai/home>
- [4] QnAMaker [Consultat a data 23/04/2019]: <https://www.qnamaker.ai/>
- [5] Qué es una KB? [Consultat a data 11/06/2019]: <https://docs.microsoft.com/es-es/azure/cognitive-services/qnamaker/concepts/knowledge-base>
- [6] Test de Turing [Consultat a data 14/06/2019]: [https://ca.wikipedia.org/wiki/Test\\_de\\_Turing](https://ca.wikipedia.org/wiki/Test_de_Turing)
- [7] Mitsuku xat de l'últim Loebner Prize Turing Test [Consultat a data 11/06/2019]: <http://www.square-bear.co.uk/mitsuku/loebner2018.htm>
- [8] Mitsuku ChatBot [Consultat a data 11/06/2019]: <http://www.square-bear.co.uk/mitsuku/home.htm>
- [9] Azure Bot Service [Consultat a data 08/03/2019]: <https://docs.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-4.0>
- [10] Conceptes bàsics de LUIS: Intents [Consultat a data 24/02/2019]: <https://docs.microsoft.com/es-es/azure/cognitive-services/luis/luis-concept-intent>
- [11] Conceptes bàsics de LUIS: Entities [Consultat a data 24/02/2019]: <https://docs.microsoft.com/es-es/azure/cognitive-services/luis/luis-concept-entity-types>